AFRL-RI-RS-TR-2011-224

INTERNET DEPLOYMENT OF DOMAIN NAME SYSTEM (DNS) SECURITY

SPARTA, INC

SEPTEMBER 2011

FINALTECHNICAL REPORT

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

STINFO COPY

AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE

■ **AIR FORCE MATERIEL COMMAND** ■ **UNITED STATES AIR FORCE** ■ **ROME, NY 13441**

NOTICE AND SIGNATURE PAGE

Using Government drawings, specifications, or other data included in this document for any purpose other than Government procurement does not in any way obligate the U.S. Government. The fact that the Government formulated or supplied the drawings, specifications, or other data does not license the holder or any other person or corporation; or convey any rights or permission to manufacture, use, or sell any patented invention that may relate to them.

This report was cleared for public release by the 88th ABW, Wright-Patterson AFB Public Affairs Office and is available to the general public, including foreign nationals. Copies may be obtained from the Defense Technical Information Center (DTIC) (http://www.dtic.mil).

AFRL-RI-RS-TR-2011-224 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

FOR THE DIRECTOR:

/s/
FRANK H. BORN
Work Unit Manager

/s/
WARREN H. DEBANY JR,Technical Advisor
Information Exploitation & Operations Division
Information Directorate

This report is published in the interest of scientific and technical information exchange, and its publication does not constitute the Government's approval or disapproval of its ideas or findings.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
SEP 2011	FINAL TECHNICAL REPORT	AUG 2004 – MAR 2011

4. TITLE AND SUBTITLE

INTERNET DEPLOYMENT OF DOMAIN NAME SYSTEM (DNS) SECURITY

5a. CONTRACT NUMBER
FA8750-04-C-0229

5b. GRANT NUMBER
N/A

5c. PROGRAM ELEMENT NUMBER
62301E

6. AUTHOR(S)

George R Mundy

5d. PROJECT NUMBER
DHSA

5e. TASK NUMBER
PA

5f. WORK UNIT NUMBER
RT

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
SPARTA, Inc
25531 Commercentre Dr
Lake Forest CA 92630-8873

8. PERFORMING ORGANIZATION REPORT NUMBER
N/A

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Air Force Research Laboratory/RIGA
525 Brooks Road
Rome NY 13441-4505

10. SPONSOR/MONITOR'S ACRONYM(S)
AFRL/RI

11. SPONSORING/MONITORING AGENCY REPORT NUMBER
AFRL-RI-RS-TR-2011-224

12. DISTRIBUTION AVAILABILITY STATEMENT
Approved for Public Release; Distribution Unlimited. PA# 88ABW-2011-4864
Date Cleared: 12 Sep 2011

13. SUPPLEMENTARY NOTES

14. ABSTRACT
This report describes the effort performed under contract number FA8750-04-C-0229 for supporting the Internet Deployment of Domain Name System Security (DNSSEC). This effort included various DNSSEC deployment and outreach activities, software development in support of DNSSEC deployment, and tasks related to managing the hardware, software and network infrastructure for the DNSSEC Deployment Initiative.

15. SUBJECT TERMS
Domain Name System, Domain Name System Security Extensions (DNSSEC), Internet Addressing

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			FRANK H. BORN
U	U	U	UU	28	19b. TELEPHONE NUMBER (*Include area code*) N/A

Table of Contents

1 Summary

Domain Name System Security (DNSSEC) was developed to counter certain specific threats in the Domain Name System (DNS) protocol. While the benefits of DNS Security are self evident, the level of adoption of this technology had been modest for a variety reasons. The Internet Deployment of DNS Security project was aimed at facilitating the deployment of DNSSEC across the wide spectrum of users.

Work on this project has focused on three main fronts: 1) standards advancement 2) outreach activities, and 3) creation of software tools and applications. This work also included managing and operating the infrastructure necessary to host various DNSSEC deployment resources within a Certified and Accredited environment.

Standards Advancement. The core DNS specifications, although very near completion at the start of this effort, required various refinements to reflect lessons learned, and to address new requirements. SPARTA has worked closely with the IETF community to work on fixes to the DNSSEC protocol and to develop operator guidance where necessary.

Outreach Activities. SPARTA has provided the overall co-leadership for DNSSEC Deployment project. We have worked closely with operators, registrars and registries in order to identify and overcome their unique barriers for deployment. We have also collaborated with Industry and have developed a number of deployment resources that serve a vast audience. We have conducted various workshops and training sessions in order to assist various operator groups in understanding and deploying DNSSEC.

Software Tools and Applications. Most of the software development work performed as part of this contract is packaged and distributed as the DNSSEC-Tools suite. This tool suite includes a number of resources for zone administrators, name server administrators and end-system administrators to enable them to easily deploy DNSSEC in their operations.

Certification and Accreditation Activities. As part of making available a number of DNSSEC-related resources to the community, and to encourage collaboration and synergy between various entities around the world, this work also includes multiple websites containing a blog and wiki engine, and a number of mailing lists. During this effort we designed and developed the Resources for the DNSSEC Initiative (RDI) system, outlined processes, developed documentation, migrated content and mailing lists and performed maintenance of the websites in accordance with DHS C&A guidelines for confidentiality, integrity and availability for a moderate confidentiality system.

2 Introduction

The Domain Name System (DNS) is a hierarchical, replicated, and distributed database that creates a map between a human-memorable domain name and information about that name, for instance an IP address. It forms one of the core infrastructures that enable the Internet to be as easily accessible as it currently is. For something so ubiquitous, DNS in its simple form is extremely insecure. Packets can be intercepted and modified, clients can be betrayed by malicious or compromised systems, and caches can be corrupted causing servers to give out incorrect responses to queries. Fallout from these threats range widely to include minor inconveniences to users, substantial financial loss due to downtime, compromise of a brand from having been attacked, and compromised security information of end-users.

DNSSEC is the result of the focused effort of the security community to add security to the DNS protocol. DNS Security raises the bar on attacks against the DNS by adding assurances that allow clients to verify that data came from a trusted source (origin authentication) and that data was not modified by an un-trusted entity either through cache manipulations or man-in-the-middle attacks (integrity protection). It achieves this by generating digital signatures over the DNS information using the zone owner's private key. Signatures that cover data within a particular zone are verified using Zone Signing Key (ZSK) and Key Signing Key (KSK) public keys that are also published in the same DNS zone. The DNSSEC protocol defines a mechanism for building an authentication chain from a set of locally recognized trust points, or "trust anchors", to signed data within DNS zones. Transaction Authentication provides a measure of assurance that messages sent between the DNS servers and clients are not modified in transit. DNS Security also provides authenticated denial of existence of data, or the ability to prove conclusively that the data being queried for is not present in the zone file. Without this ability it would be possible for a malicious entity to force the originator of the query to see a spoofed response before the actual response came in with possible security ramifications.

Security extensions to the DNS were originally proposed in 1995. It has taken the better part of the last decade, amidst changing requirements, to define a distributed authorization structure that could be retrofitted into the existing DNS structure. While the benefits of DNS Security are self evident, the level of adoption of this technology was modest.

Perhaps the biggest impediment to widely deploying DNS Security was the disinterest exhibited by a large segment of the Internet community over adopting this technology. The reasons for this lack of interest were either due to lack of awareness on the part of the organizational policy makers about how vulnerable their DNS data currently was, or due to the paucity of tools and techniques that would make DNS Security operationally viable for many organizations. In many cases deploying DNS Security was a business decision because of the increased operator overhead costs and increased processing and bandwidth requirements. Growing the deployment base for DNS Security required educating the operators and policy makers about DNS Security and its operational viability.

There was also a shortage of tools to help minimize the operator overhead incurred while performing DNS Security operations. Tools not only make the life of the zone operator easier but also minimize the number of errors that can be introduced due to mis-configurations and oversight. Tools are also needed to facilitate effective debugging. There were very few test facilities and even more limited assistance available to hand-hold operators through the deployment process if and when they choose to do so. For many operators, DNSSEC represented added complexity: the increased responsibilities for zone data administrators, the need for configuring trust anchors (and ensuring that they stay current), and the need for integrating DNSSEC into the normal name registration workflow. Operator experiences from initial DNS Security deployment efforts had to be made available to future operators in order to prevent them from making the same mistakes as their predecessors. While tools can automate some of the operational overhead, in certain cases, such as emergency key rollovers, manual intervention is necessary, and thus appropriate guidance was necessary.

There was a significant lack of DNSSEC-capable software for end systems. End systems such as browsers are responsible for a large number of DNS queries in the Internet. This number will only increase as the number of intelligent end devices that are DNS Security aware also increases. End-users have a choice on the level of trust they place on the network they are connected to and the amount of DNS Security work they are willing to perform. If the name server being connected to is sufficiently trusted, then the user can rely on this name server to relay the result of its validation checks to it. Conversely, in an environment where the threat is higher, the end-system may do most of the cryptographic validation itself. This is especially true in a mobile environment where the user does not have the benefit of a secure enclave setting and has no known secure path to a DNS Security server. Unless the user knows how to interpret the results from a DNS Security query she will not be able to gauge the verity of the domain contents she is accessing. The problem not only deals with identifying the exact error result but also extends to communicating this value to the end user in an understandable format. At the start of this effort there were no good examples of end systems that were able to fetch, understand and act upon DNS Security validation results.

The goal of the Internet Deployment of DNS Security was to increase deployment through the accomplishment of three main tasks. The first task was to follow up on the completion of essential DNS Security protocol specifications and Best Current Practices, and performing additional research in the areas of key management and end user interaction with the DNS Security infrastructure. The second was that of growing the deployment base of DNS Security by way of education and outreach, both nationally as well as internationally, across government and commercial entities. The task also entailed provisioning of hands-on workshops, training sessions, detailed operational guidance documentation and test facilities that hand-held the operators through the initial and subsequent phases of DNS Security deployment. The third task was providing essential software and tools that assist operators in performing routine and emergency DNS Security operations with minimal overhead.

The remainder of this document discusses the different activities performed in support of the Internet Deployment of DNS Security.

3 Methods, Assumptions and Procedures

This section describes the various contributions that we have made on various fronts as part of the Internet Deployment of DNSSEC project.

3.1 Contributions towards Advancement of Standards

At the start of this effort the DNSSEC specifications had been published as Proposed Standards, which is the first step of a three-step process for Internet Standards. However various refinements were required to the protocol in order to reflect new advances in cryptography and to address new requirements as deployment was extended to other ccTLDs.

DNSSEC was primarily focused on providing mechanisms that allowed a validator to ensure that an answer received was the same as the record placed in the zone. Confidentiality of data was not a concern. As DNSSEC deployment progressed, certain registries observed that the mechanism that was used to prove the non-existence of name in DNSSEC violated their privacy laws by allowing any resolver to "walk" the zone. This led to the creation of a modified version of NSEC (National Secure Record), namely NSEC3. SPARTA was closely involved in various design discussions related to NSEC3. We participated in multiple NSEC3 workshops in order to validate various assertions made by the protocol developers and to test interoperability among several implementations. At this time we also added NSEC3 support to our validator library. In the shadows of the NSEC3 discussion was also born the approach of online signing using "epsilon" signatures, which we were actively involved in. Online signing, though not very popular, may prove useful in certain use-cases as deployment progresses.

SPARTA has authored, commented on and improved a number of Internet drafts. We have put together a number of technical papers such as a Statement of Needed Capability on Trust Anchor Repositories (TAR), the initial Root signing specification, and Root key publication alternatives. We are also credited in contributing to a number of Request For Comments (RFCs) including the base DNSSEC specification, and have led numerous sessions at the IETF to try and stimulate more discussion on the topic of DNSSEC capability in applications and Trust Anchor Repositories.

The following lists some of the Internet Drafts and RFCs that have been authored by SPARTA as part of this contract.

3.1.1 DNSSEC-bis document

This document is a collection of minor technical clarifications to the DNSSEC document set. It is meant to serve as a resource to implementors as well as an interim repository of possible DNSSEC errata.

3.1.2 DLV

DNSSEC Lookaside Validation (DLV) is a mechanism for publishing DNSSEC trust anchors outside of the DNS delegation chain. It allows resolvers to validate DNSSEC-signed data from zones whose ancestors either aren't signed or refuse to publish Delegation Signer (DS) records for their children.

3.1.3 Using SHA-256 in DS Records

This document specifies how to use the SHA-256 digest type in DNS DS Resource Records (RRs). DS records, when stored in a parent zone, point to DNSKEYs in a child zone.

3.1.4 DNSSEC Validator API

The DNS Security Extensions (DNSSEC) provide origin authentication and integrity of DNS data. However, the current resolver Application Programming Interface (API) does not allow a security-aware resolver to communicate detailed results of DNSSEC processing back to the application. This document describes an API between applications and a validating security-aware stub resolver that allows applications to control the validation process and obtain results of DNSSEC processing.

3.1.5 Requirements Related to Trust Anchor Rollovers

Every DNS security-aware resolver must have at least one Trust Anchor to use as the basis for validating responses from DNS signed zones. For various reasons, most DNS security-aware resolvers are expected to have several Trust Anchors. For some operations, manual monitoring and updating of Trust Anchors may be feasible, but many operations will require automated methods for updating Trust Anchors in their security-aware resolvers. This document identifies the requirements that must be met by an automated DNS Trust Anchor rollover solution for security-aware DNS resolvers.

3.1.6 Requirements for Management of Name Servers for the DNS

Management of name servers for the Domain Name System (DNS) has traditionally been done using vendor-specific monitoring, configuration and control methods. Although some service monitoring platforms can test the functionality of the DNS itself there is not an interoperable way to manage (monitor, control and configure) the internal aspects of a name server itself.

This document discusses the requirements of a management system for name servers and can be used as a shopping list of needed features for such a system.

3.2 DNSSEC Deployment and Outreach Activities

3.2.1 DNSSEC Deployment Initiative Co-leadership

SPARTA has provided overall co-leadership for DNSSEC Deployment project. We have helped identify potential problem spaces for DNSSEC deployment and have coordinated a strategy for overcoming those issues (examples include TARs, middle box issues, and algorithm agility).

We have also supported the development of the DNSSEC Roadmap by identifying various adopter scenarios that would benefit from DNSSEC, identifying DNSSEC aggregation scenarios where the costs of deployment could be distributed over a variety of subscribers, and by performing a gap analysis of available technical pieces for DNSSEC operations. As part of performing the gap analysis we also developed a DNSSEC software tracker [SFT_TRK] that serves as an online list of software components that are currently available to support DNSSEC operations.

SPARTA was closely associated with the initial signed Root zone design activity and published the initial list of deployment alternatives for a signed Root zone. We participated in the Root signing symposium organized by the Public Interest Registry (PIR) and the DNSSEC coalition, where we identified the set of actions that were needed for successful deployment at the Root beyond the publication of a signed zone. We also worked on defining an operational model for Trust Anchor Repositories, which arguably also hastened the pace of deployment at the Root zone.

SPARTA has provided guidance to the .mil operators to help them with their initial DNSSEC deployment strategy and has given an overview of DNSSEC to the California CISO as part of their effort to overhaul their DNS infrastructure. We worked closely with PIR in its DNSSEC provisioning testing for the ORG zone, and also worked closely with certain registrars (DynDNS and NamesBeyond) in adding and testing DNSSEC capability to their products. We worked closely with Shinkuro in testing a "ripple free" DNSSEC operator transfer process, wherein transfer of the DNSSEC service would not impact the zone's availability with respect to DNSSEC and co-authored with Shinkuro Inc a "DNSSEC crib-sheet" - a document meant to assist registrars in specifying meaningful defaults for various DNSSEC parameters while configuring their systems. We have also engaged with ccTLDs and industry in order to share our experiences with DNSSEC deployment in various scenarios, on simplifying DNSSEC operations through the use of tools, and in enabling DNSSEC validation on end-systems. We have presented and demonstrated tools at various conferences and have published our work in various peer-reviewed journals.

SPARTA has also led the DNSSEC workshops at ICANN meetings held throughout the world. The focus of these meetings is to continue to highlight the value of DNSSEC, to involve the local DNS community in the DNSSEC Deployment effort, and to gather important feedback on specific deployment use-cases.

We teamed with the National Institute of Standards and Technology (NIST) in conducting DNSSEC "Policy to practice" workshops targeted towards DNS operators. We have also organized DNSSEC sessions, prepared handouts, and presented in multiple FOSE/GOVSEC conferences.

3.2.2 Step-by-Step Guides

Administration of a DNSSEC-signed zone is more complex than that of an unsigned zone. Care must be taken to keep keys and signatures current, and not let them expire. If the zone is compromised, then the zone's administrators must take action to restore the zone's place in the DNSSEC chain of trust. Unless the normal and emergency DNSSEC administrative operations are performed properly, a zone and its chain of trust may be put at risk.

The Step-by-Step Guide [DT-SBS] was written to assist zone administrators in gaining operational experience with DNSSEC. It provides lists of instructions for a number of normal operations situations encountered by zone administrators: key creation, signing non-delegating zones, signing delegating zones, and rolling over keys. In addition, it provides instructions for handling several emergency situations when a key has been compromised.

3.2.3 US Marine Corps Experiment

As part of an effort to deploy DNS Security in the US Military environment, we created a DNSSEC Experiment to provide exposure of the DNS Security protocols and tools to operators of DNS servers in DISA and the greater Department of Defense (DoD). While the main goal of the Experiment was to increase operator awareness and knowledge of DNS Security, an interesting outcome of the experiment was our learning how difficult DNS Security deployment was going to be in a large, diverse, and separately-managed organization such as the DoD.

One of the greatest challenges for the Experiment was getting management support to direct the operators to spend time working with the Experiment. Part of this stemmed from the difficulty in educating management and helping them to understand the problems and solutions.

Another challenge, although not surprising in nature, was the education of the operators. As is the case with DNS operators Internet-wide, the operators involved in the Experiment had a varying degree of knowledge and understanding of the DNS. This variance required different levels of effort for the education of each operator.

3.2.4 DNSSEC Split views

Split-view DNS is the term used to describe multiple views of DNS information for a domain based on where and by whom the query is sent. Split-views help contain DNS names to only those portions of the network that need to see these names. Although primarily meant to be a network management technique, the tailoring of the DNS to create an internal view of information hidden from the outside is also seen by some as improving their organization's security posture, by preventing the exposure of internal host names, knowledge of whose existence is deemed to be sensitive. Relying solely on split-view DNS to protect sensitive hosts from attacks has proven to be less than adequate in the past. Attack vectors in recent Internet exploits have been able to successfully infect hosts with or without their IP addresses being published in the DNS. Conversely, publishing the IP addresses of hosts that are otherwise secured does not necessarily increase their vulnerability to these attacks. Name hiding through split-view DNS is primarily useful as part of a more comprehensive defense-in-depth strategy to provide one line of defense against name-based attacks.

DNSSEC has some unique problems in a split-view environment. Origin authenticity and data integrity are determined by validating the chain-of-trust from the signed record to some trusted key configured at the end resolver. In the case of split-view DNS every chain-of-trust in every view must validate properly. Some names may be common between multiple views but contain different data. Cache pollution is a possibility when data from the wrong view is returned in response to a query. Building a chain-of-trust from a trusted key above the zone that has split views, to data in the internal view of a zone can be especially problematic, caching problems notwithstanding.

SPARTA developed a document that provides recommendations for correctly configuring the split-view DNSSEC environment in a typical enterprise network. This document is important from two angles: it draws attention to the fact that split-view DNS is not the most efficient way to hide names in the DNS. It also provides guidelines for enterprises on how to avoid basic problems while configuring this setup if they chose to use this approach for name hiding. The objective in either case is to ensure that split-view DNS does not prove to be a deployment show-stopper for those organizations that heavily rely on split-views for their normal operation.

3.2.5 DNSSEC Crib Sheet

This memo [CRIB-SHEET] provides advice on the values to choose for the configuration associated with DNSSEC that provide good security without causing an undue burden on operators' name service infrastructures. The configuration parameters include key sizes and lifetimes, re-signing periods, and time-to-live (TTL) for the records.

3.3 DNSSEC Software Development

At the beginning of the project, the project team members held multiple conversations to map-out both the details of the DNSSEC deployment problem space to be tackled as well as the potential tools that could be developed to increase the likelihood of successful DNSSEC deployment. Two categories were identified as the most crucial to examine: zone operators and end-users of DNSSEC. In order to achieve successful deployment, both of these categories had to have easy-to-use solutions to maximize the rate of deployment. Once these critical categories had been discussed, the project team further examined what tools existed already that solved some of the problems in these two categories. The project team then discussed the details of both of these categories to determine which missing components would provide the greatest impact in future deployment. The team decided an important goal was to produce tool solutions that filled the most immediate needs of operators and end-users. This was combined with a secondary, but important, emphasis on creating user interfaces that were as easy as possible to use so that deploying and utilizing DNSSEC would not be a significant burden.

Since the tools were to be used in a number of highly diverse environments (most operators claim that their operation is "unique" in some way), the project team decided to make the tools as modular and extensible as possible. Modular tool development also improved the team's ability to roll out the tools incrementally. Multiple incremental software releases were made allowing project components to be distributed quickly so that feedback could be gained from industry experts early on. As the tools received feedback, the results and advice were incorporated into the software for the next release.

The next step was to actually produce the tools that the team had agreed upon. In order to achieve this, the team utilized a number of techniques to maximize productivity. The team was highly distributed, with team members working in multiple locations across the country. To ensure that the distributed nature of the group was efficient, the project team used a number of communication mechanisms such as E-mail, phone conferences, and jabber chat rooms. The work was divided into modular sections and each piece to be developed was assigned to an individual developer or set of developers. The various tasks were prioritized based on the difficulty and desirability of the results. More pressing needs were generally fulfilled first, especially if the time required to implement the results was small. The team registered a domain name for the project (dnssec-tools.org), and initially used SourceForge's web, code repository, mail, and other project management features before migrating these pieces to in-house machinery.

We maintained three mailing lists for the project: dnssec-tools-users, dnssec-tools-bugs, dnssec-tools-commit. We also set up a bug-tracking database to track feature and bug requests submitted by the dnssec-tools user community.

The following sections describe some of the design and engineering choices made in evolving the technology developed as part of this effort. Further details of individual libraries and tools

that were developed part of this effort are discussed in other contract deliverables [See Section 8.1].

3.3.1 Server-end Tools

In order to prepare a DNS zone for DNSSEC validation and name resolution, encryption keys must be generated and the zone must be cryptographically signed. The BIND [ISC_BIND] software includes easy-to-use tools to perform these actions. However, for proper DNSSEC zone administration, a fair bit of record-keeping is required in order to keep track of current and expired encryption keys.

The zonesigner command was developed to assist with the tasks of key generation, zone signing, and record keeping. It acts as a front-end to the BIND distribution's dnssec-keygen and dnssec-signzone commands, which perform the actual work of generating keys and signing zones. zonesigner manages a database of information, called keyrec files, that records how the keys were generated and how the zones were signed.

Keyrec files record information about key generation and signed zones. They contain key-specific records and zone-specific records. The key records describe how the key was generated. This information includes encryption algorithm, key length, key type (KSK or ZSK), and key-generation date. Zone records describe how a zone was signed. This information includes signing keys, expiration time, and zone-signing date. As new keys are generated, new keyrec records are created and added to the keyrec file. When a zone is first signed, a new zone record is added to the file. As key roll-over occurs for a zone, the zone's keyrec record is updated to reflect the new signing. By maintaining the parameters for key generation and zone signing, zonesigner can automatically generate new keys and re-sign the zone in the previous zone signing cycle.

The Step-by-step Guide [DT-SBS] was used as a rough design document. The Guide's sections on signing zones were used to determine the steps and their proper order to correctly sign a zone. The normal key roll-over sections of the Guide were used to determine the steps needed to implement key roll-over in zonesigner.

zonesigner automatically takes the following steps when signing a zone:

- generate a KSK key
- generate several ZSK keys
- copy the zone file
- update the zone file's serial number
- add include lines for the keys
- sign the zone file

In contrast, this operation would have taken many additional steps to accomplish using native BIND utilities.

Once we developed zonesigner, we quickly realized that we needed a separate tool to automate the management of ZSK and KSK rollover operations for DNSSEC zones. The goal was to enable operators to be able to generate new keys in a periodic frequency, and be able to do so without causing disruptions on the network as a result of cached data existing in external recursive resolvers. We implemented this functionality in the rollerd utility, which was capable of being run as a daemon program or in a stand-alone "single run" mode. Program modularity was a goal, so most of the tools were comprised of smaller modules with rollerd internally invoking zonesigner to do most of the zone signing operations.

Network administrators always test their network infrastructure before it is deployed in live environments. DNSSEC-enabled DNS zones will fall into this category as well. Thus, it is important that before the role out of a DNSSEC enabled zone is to occur operators will need a tool to check that the DNS data is in fact properly secure and meets the criteria imposed by the DNSSEC specifications.

To achieve this goal, a DNS data validation program called "donuts" was developed. This tool is a highly extensible tool that loads in a number of rule definition files, and at least one DNS zone data file. It then executes each rule against the zone file's data and reports any results to the operator. The rules that come with the application are currently derived mostly from the DNSSEC specifications, although a few more general DNS rules were added as well. If the zone data file being analyzed contains errors like cryptographic signatures that do not match the data, expiration dates that are past, sub-domain delegation records and cryptographic signatures which do not match the deployed live data, etc, then donuts will notify zone operators of those conditions.

In addition to this tool, a long-running daemon ("donutsd") was also developed that runs donuts on periodic intervals and e-mails the results to zone administrators. The combination of these tools lets zone administrators test their DNS data prior to deployment, as well as monitor the continuous health of the zone data as it is affected by the passage of time.

The donuts tool was written by first designing a rule definition syntax so that rule files could be loaded at run-time. This allows for 3rd party extensibility and lets network administrators define their own additional rules that their own network's zone data may need to conform to. Once the rule syntax was established, the internet-drafts that currently made up the DNSSEC specifications were examined to come up with rules that checked for proper DNSSEC zone data. These rules were combined with rules of a best-practice nature that operators would likely want to follow as well.

In order to be as usable as possible in different operational environments, the tool is highly flexible allowing not only for additional, potentially local, rules to be executed on zone data but also to allow for enabling and disabling of individual rules. Finally, configuration files allow local tuning of rule parameters to meet the requirements of local environments. This flexibility

ensures that operators can fine-tune the output of donuts to meet their particular zone requirements.

One of the difficulties in managing any complex set of data is getting a conceptual grasp of the entire dataset as a whole. Data visualization techniques can often aid in helping administrators better understand their network and related infrastructure. The mapper program, developed early on by the project team, takes DNS zone files and plots visual representations of them. These representations are useful to study in order to see if the real zone data that was created by an operator matches their expectations. As a case in point, before donuts was created later in the project, the results of the mapper program visually flagged some errors within some of the project's test zones. The test-zone operators were not aware of these errors as they had only been able to study the zone's data files themselves. It was the visual representation that allowed the team to catch the deployment error.

The tool was developed by tying together an existing perl module that is capable of parsing zone files together with another perl module that can generate graphviz [GRAPHVIZ] diagrams. Each type of DNS record is then mapped to different colors, line types or shape types to allow distinctive visual patters to be represented in the resulting diagrams. The program is highly flexible, allowing administrators to hand-tailor the output of mapper to fit their specific needs by use of data filtering and personalized coloring, shaping etc.

3.3.2 Resolver-end Tools

The interactions of the DNS protocol are very complex to analyze and follow. The number of hosts involved with even a simple DNS query can range anywhere from 1 host with a lot of cached data to as many as 10 hosts when very little data is currently available in the resolver's cache. More complex queries can involve the cooperation of ever more DNS servers. When debugging DNS queries, looking at the output of packet traces is often helpful but still difficult to analyze due to the large number of queries that are frequently involved in satisfying a request. The additional data elements and protocol semantics that DNSSEC adds to the complexity of normal DNS queries make it even more difficult to debug DNS queries when DNSSEC has been enabled on a network. However, it is certain that operators and application developers will need to perform rigorous analysis of DNSSEC-enabled software before they are willing to deploy it in live environments. Once deployed, when problems arise they will need tools to quickly analyze the situation.

The dnspktflow tool, which was developed during this project, helps operators analyze captured packet traces in order to get a better understanding of the captured queries and responses. By showing the results graphically in a flow diagram, it is easier to visually see the direct impact that queries had upon the DNS servers and clients within a network. It carefully labels flow traces with information which is critical to understanding DNSSEC packet flows. The result is an image, or series of images, that shows the packet-by-packet flow of DNS packets from

captured network traffic. The resulting visual representation allows operators to quickly inspect recent traffic to look for DNS traffic problems within a network.

The tool was developed by capturing the output of the tethereal [TETHER] network traffic analyzer and feeding the parsed results into the GraphViz perl module. This produces the visual diagrams of DNS packet traces. More information was then parsed and added to the flows visual to show the DNSSEC specific details of the information within these packet flows.

One of our goals for this project was to make logging of DNSSEC activity more meaningful by supplying a mechanism for parsing the sometimes voluminous logs generated by BIND 9.x. We chose logwatch [LOGWATCH] for its simplicity, widespread deployment, and easy customization. Logwatch is organized as a set of scripts and configuration files for each type of log file you wish to analyze. These allow Logwatch to parse the log files specified, group specific types of entries, and present them all in one file for the system administrator to view. This project developed configuration files and scripts extending logwatch to enable analysis of DNSSEC-specific log messages.

Trust anchor management is an important component of managing a validating resolver. SPARTA was instrumental in developing the requirements that led to the adoption of the RFC 5011 “timers” approach for automated trust anchor rollover. Our team also built Trustman, the first automated trust anchor rollover implementation that detected and rolled over trust-anchors in accordance with RFC 5011. Since different validator implementations supported different trust anchor file formats, we also developed the convertar utility that enabled easy conversions between different formats.

3.3.3 Applications and Libraries

The previous section described tools developed by this project that were mostly targeted towards network operators. While these tools are important for DNSSEC deployment, it is also essential that popular Internet applications be made DNSEC-aware, so that the benefits of DNS security become available to end users. Making applications DNSSEC-aware (rather than just making the underlying infrastructure DNSSEC-aware) will make it possible for applications to provide configuration options to users regarding DNSSEC validation. This will give users more control over the results of DNSSEC validation.

This project has developed patches to various popular network applications to make them DNSSEC-aware. These applications were chosen based on various criteria including how widely the applications were being used, the availability of source code, a friendly license, and active developer community. These applications are further described in the project web page [DNSSEC-TOOLS].

Central to validation within applications is the DNSSEC validator library and the accompanying resolver library. These libraries provide the resolver and validating components for DNSEC

validation. They provide interfaces to fetch answers from a DNSSEC-aware name server and basic functionality for resource-record validation. Most of the implementation of the resolver component and some for the validator library was re-used from an older implementation of the secure resolver library built per the RFC 2535 DNSSEC specifications. Modifications were made so that our library was conformant to the newer specifications and also contained other feature improvements. The choice to split the secure resolver functionality into two libraries was made in order to decouple these two distinct roles. Given a well-documented interface between these two components it could then be possible to plug the validator component with applications that contain their own native implementation of the resolver functionality.

Our initial library only contained synchronous versions of the query lookup and validate routines. During the course of adding DNSSEC capability within applications, we realized that an asynchronous interface was also highly desirable. Therefore we also added in this capability.

We realized early that obtaining conformance on a validator API would be important as deployment in end-applications grew. This project has developed a functionally rich Application Programming Interface (API) for the validator library. The API functions provide a convenient way for applications to control the DNSSEC validation process, and to obtain results of validation.

The API can be broadly divided into three groups: high-level application interface, core validation-check interface, and validator policy interface. The high level functions, on the other hand, are designed for ease of use, and mirror existing DNS-related functions. They provide less control over the validation process and return a consolidated validation status. They are best suited for existing applications that already use legacy DNS-related functions such as gethostbyname(), getaddrinfo() and res_query(). The core validation-check and validator policy interfaces provide more control over the validator policies and configuration, and return detailed validation status information. They are best suited for sophisticated applications that need more control and insight into the validation process.

As part of the validator library, we also developed the validate command line tool. This tool outputs the results of DNS query resolution and DNSSEC validation using the functions provided by libval. This tool is primarily designed to aid operators in diagnosing DNSSEC problems. We also equipped this utility with a graphing utility drawvalmap that allowed a user to inspect the different validation chains created by the validator, and more importantly to identify why a particular query did not validate.

In order to test our validator library against various corner cases, we also developed an online test zone. The test zone has about 365 names to test against and includes various cases of DNSSEC brokenness. Its usefulness beyond simply being that of a libval testsuite became obvious immediately, so we created the maketestzone utility that vendors can use to customize their own unique test cases. The test zone has been an important resource for the resolver community and a number of different vendors have used our test zone to look for problems in their implementation.

3.4 Certification and Accreditation Milestones

We initially used sourceforge [SRCFRG] as our main hosting service for our code repository and our development mailing lists. With the move by DHS to certify and accredit all systems that it is responsible for, we moved the code repository server, web server and mail servers in-house. The dnssec-deployment site [DNSSEC-DEPLOY] was also included in this certified and accredited environment.

The C&A related tasks comprised the following

- Understanding the different tasks and the compliance office's requirements
- Defining the system boundary and roles.
- Understanding the existing processes in place at the contractor site (SPARTA).
- Definition of processes such as daily, weekly, monthly backup operations, procedures for reviewing changes, and audit log review.
- Configuration of hardware and software.
- Development of custom scripts to help manage certain processes, checklist of continuing operations.
- Table-top testing of our Contingency Plan and Incident Response plans.

As part of this process we submitted the following documents:

- E-authentication worksheet
- FIPS-199 worksheet
- Risk Assessment & RA Observations
- System Security Plan
- Configuration Management Plan
- Configuration & Maintenance Guide
- System Rules of Behavior (ROB) and ROB Training Slides
- Access Control Policy
- Contingency Plan
- Incident Response Plan
- List of system exceptions

The RDI system was initially classified as a “low” confidentiality system since it only processed information in the public domain and maintained open mailing lists. Once we completed the initial set of documents for the C&A, the system confidentiality level was revised to be “moderate”, which meant that most documents had to be updated. We had to perform a revision of our system design when the system was reviewed against DHS S&T Enterprise Architecture. The system was finally reconfigured to meet all outstanding C&A requirements.

4 Results and Discussion

The results and accomplishments from this effort are documented in various technical reports submitted at various times during the project. A summary of different reports for the different task areas is provided below.

4.1 Standards Advancement

The complete specification for the DNS system is spread across a large number of documents, with no clear guidance on an orderly progression through them. The DNSSEC System Specification documents provide this guidance, listing the DNS-related documents in relation to the following "flows":

- Lookup Flow
- Registration Flow
- Application Flow
- Operations Flow

4.2 DNSSEC-Deployment for various classes of adopters

The DNSSEC Security Deployment Plan identifies the building blocks required to support widespread deployment of DNSSEC. It does so by providing five roadmaps in the following areas:

- Operations
- Software
- Management, measurement, and evaluation
- Communications and outreach
- Issues

The report DNSSEC Awareness in Applications focused on DNSSEC-awareness in applications, and ways in which applications could expect to use DNSSEC services.

We also summarized our results for the base and extended USMC DNSSEC Experiments.

4.3 Software Developed

Software developed under this contract is packaged as the DNSSEC-Tools suite [DNSSEC-TOOLS]. We have developed a number of utilities to ease zone administration and resolver administration for DNSSEC. We built the first end-system validator and the first implementation of RFC 5011 for Trust Anchor management. Currently we have over 100 subscribers on the dnssec-tools-users mailing list, many of which are using our tools. Certain users have used

components of the SPARTA tools within their own commercial products. SPARTA has also been engaging with various commercial software and OS vendors in order to extend the benefits of validation to a wide user base.

The project has strived to make frequent updates to its code-base and to provide these updates to its user base on a timely basis. The deliverables provide pointers to software releases at various times in our contract.

We have delivered the software documentation under the following deliverables:

- The Software User's Manual describes the software developed for the DNSSEC-Tools project. It contains installation instructions for the software, as well as man pages for the commands and library routines. Additionally, the Step-by-Step guide assists system operators in gaining familiarity with DNSSEC operations. It discusses key generation, zone signing, key roll-overs, emergency key roll-overs, and serving signed zones. Information is provided for several types of zone configuration. We also created a document titled "Adopter Scenarios for DNSSEC-Tools", where we look at various ways in which the DNSSEC-Tools could be tailored to serve different zone operator and resolver operator needs.
- For later versions of the DNSSEC-Tools suite we moved all documentation to the wiki in order to accommodate more timely updates to the document, and incorporate feedback from our users. The use of this mode of the software users manual was described in a report.
- Another report provides the final summary on some of the tools and applications available in the DNSSEC-Tools package for use by the DNSSEC data providers (zone administrators and authoritative server administrators) and the DNSSEC data consumers (recursive server administrators, end-system administrators and users). These include tools for operators of all networking types ranging from DNS content producers to DNS content consumers.

4.4 Certification & Accreditation

The System Security Plan, the Configuration Management Plan, the Configuration & Maintenance Guide and the Contingency Plan have been submitted.

5 Lessons Learned

Some of the lessons that we learned during the completion of various task pieces associated with project are listed below.

Advancement of Protocols and Standards

- Although the DNSSEC protocol was considered nearly complete at the commencement of work on this project, fairly substantial improvements were required to meet the requirements of different user groups. The NSEC3 effort and the Automated Trust Anchor Rollover mechanism are two examples of changes that SPARTA specifically contributed towards the development of.
- While we have done considerable work on understanding the different adopter scenarios for DNSSEC, there will still be cases where we will have to give special consideration and tailor our tools accordingly to meet those special needs.
- The protocol extensions that may be required for the last-mile are still not completely defined. This is partly because it is unclear whether standardized approaches are really required or which approach to adopt in particular. There will certainly be more protocol and standards related work as deployment grows to include diverse forms of end-systems.

DNSSEC Deployment and Outreach

- Zone operators have highly diverse requirements thus highly diverse requirements for the tools they need.
- Zone operators are paranoid about their zones since when DNS breaks everything does.
  - they need tools that they feel are trustable
  - they need tools that help them test their zones beforehand
  - they need tools that help them troubleshoot live environments
- It is expected that just security policies in networks today differ from location to location that DNSSEC security requirements will also differ from location to location. Roaming users will be most affected by change in policy.
- The most important lesson learned so far from the U.S. Marine Corps Experiment is that the deployment process of DNS Security is going to need to be tailored to a wide array of DNS environments. This tailoring, while most likely deriving from a group of generic "Deployment Approaches", will more than likely require specific tailoring for each and every environment.
- Another major lesson still being learned from the U.S. Marine Corps Experiment, as outlined in section 4.2. above, is that management and operator understanding is critical to successfully deploying DNS Security. Getting management to understand the problems and solutions of DNS Security and then having them allocate enough resources (operators, engineering, hardware and/or software) to deploy is hard. Additionally, educating operators on how DNS Security works and how it will integrate into their DNS environment is hard.

Software Development

- The amount of work needed to enable DNSSEC within an application greatly depends on the design of the application. Well designed applications that account for future potential changes to lower level APIs, such as DNS queries, are well set up to require only minimal changes to the application to enable DNSSEC. However, applications that handle each DNS error individually in multiple places within a code base will take a lot more work to enable DNSSEC functionality within them.

- Getting as much early and frequent feedback as possible helps ensure that tools that are developed will meet the requirements
- In order to achieve faster DNSSEC deployment all of tools, education, outreach and standardization efforts are needed. Concentrating on just one of these topics will not achieve sufficient improvements to deployment speed.
- The use of good coding practices and the development of modular tools helps in making the development effort more efficient. Incremental and modular development means faster rollout of usable tools.
- Including the community in various design decisions, as we have done through our mailing lists and wiki, helps in getting early feedback on what changes the user really desires and what changes will break their operations. Timely resolution of any problems that the users report is extremely important.

Certification and Accreditation

- Obtaining a good assessment of the system classification level is extremely important early in the system life-cycle, since this optimizes effort.
- Using an automated patch update policy for standard applications, especially for a low availability system, is very useful in keeping the system continuously protected against evolving threats.

6 Conclusions

DNSSEC is complex, but the availability of good tools can greatly reduce its complexity. It is commonly perceived that the complexity has been one of the primary reasons for the slow uptake in deployment of the DNSSEC protocol. The DNSSEC-Tools package provides a large quantity of tools that greatly simplifies the process of administrating DNSSEC in both provider and consumer environments. The DNSSEC-Tools package has also served as a role model for other open-source and commercial DNSSEC solutions and has shown other implementers how to successfully simplify DNSSEC use by DNS producers and consumers.

The list of tools and software for DNSSEC continues to grow impressively. The DNSSEC-Tools project, in particular, provides a number of open source tools for the provisioning and the consumer end of DNSSEC-enabled zones. This software suite includes software patches for a number of well-known Internet applications in order to make them DNSSEC capable. Various other open source tools for DNSSEC are also freely available in the community. While more work is still needed to fill in some of the remaining useful software components, the DNSSEC software that is available today provides a solid foundation of tools needed for administrators, developers and end users to start deploying DNSSEC today.

Finally, although a large amount of functionality has been implemented in end-user applications and services, such as in web browsers, e-mail readers and SMTP servers many more applications should be looked at. Although sometimes significant work is needed to update an application so that it becomes DNSSEC aware, there is significant added benefit once the work is completed. Applications not only offer additional security features but can also offer more decisive messages to return to the end-user.

End-user policy and validation is more complex than the tools required to produce DNSSEC valid zones. This is primarily due to the fact that DNSSEC will not be deployed in all zones on the Internet instantaneously. Because of this end-users will have to make policy decisions about which zones are required to be secure versus those that acceptable even if they're not securely deployed. The DNSSEC validation library that has been developed during the course of the project has been highly effective as a base upon which applications can depend upon for DNSSEC validation. However, these more complex policy decisions could be made and the applications need to be extended further to make use of them. The benefits to in-application DNSSEC validation, particularly with respect to enabling fine-grained local control and providing better error messages to the end-user, far outweigh the costs. The distribution and management complexity of such a system is certainly higher. However, the benefit of a completely secure deployment regardless of location and the added security and usability benefits provided to the application significantly outweigh the required complexity increase.

7 Directions for Future Work

DNSSEC has progressed beyond being an obscure technology to one that is being seriously considered in various operational zones. The next step should be to focus on making DNSSEC an integral part of the way users operate with networked systems.

The zone administration tools developed within the project greatly reduce the burden of administrators for developing and maintaining their DNSSEC-enabled zones. As a next step the tools should be integrated into normal operator workflow/standard front-ends so that operators do not have to think about managing DNSSEC any differently than they do for DNS. The project team has also envisioned, but not yet implemented, a tool "suite" which would wrap the existing project and external tools together in a easy-to-use interface which could help hand-walk a new administrator through the process of deploying a secure zone. The project team designed their tools to be independent in nature to allow for operators of specialized environments to be able to use individual solutions. The next step in this process would be to provide a binding that integrates the compartmentalized tools into one "suite" as well.

From the consumer/end-user side goal is to have DNSSEC enabled as the default: users should expect DNSSEC availability but should not have to consciously think about it. People should not feel the need to disable DNSSEC for routine tasks. End-devices are also getting smaller and more powerful. Operating environments for mobile smart devices are seldom within the

organizations' control and often include insecure environments. That means that users accessing such devices are more likely to be subject to attacks that may bring harmful components back into the secure environments. Enabling DNSSEC in other widely-used applications, for widely-used platforms with a focus on enabling system-wide DNSSEC validation is the key to getting the benefits of validation to the end-user.

There are a number of opportunities to completely break the DNSSEC chicken-and-egg deployment model. New security protocols that will make use of DNSSEC to simplify bootstrapping (e.g. SSL/self-signed cert leap of faith) can greatly simplify operator overhead and provide added benefits to the user. There are already a number of existing protocols that will benefit from DNSSEC, such as RADIUS, and these should be looked into.

There are also a number of additional education and outreach opportunities that will move DNSSEC adoption even further along. Standardization may be required for the user-end space such as in the definition of a validator API and last mile validation, since not all applications will be DNSSEC capable, and not all end systems will run a recursive name server. Identifying the most effective way of priming Root trust anchors for applications or end-users (e.g. when vendor does not distribute the key) is another area of future work. Additional standardization may also be required for various types of parent-child or Registry-Registrar-Registrant communication, especially during KSK rollover events.

8 References

8.1 Deliverables

"DNS Security Deployment Plan: Domain Name System Security Roadmap, Software Pieces", Technical Information Report, 27 November 2006

"DNS Security System Specification", Technical Information Report, 30 November, 2006.

"Software User Manual (SUM): Training, Procedural, and Development Documentation", 27 November 2006.

"Software Product Specification", Executable Software, Source Files and Packaging Requirements, 27 November 2006.

"Base USMC Experiment Report", Technical Report, 31 May 2005

"Extended DISA/USMC DNSSEC Experiment Report", Technical Report, 31 August 2005.

"DNS Security Deployment Plan: DNSSEC Awareness in Applications", Technical Information Report, 31 Dec 2008.

"DNS Security System Specification: Current Status of DNS Security Protocols and Standards Advancement", Technical Information Report, 31 Dec 2008.

"Software User Manual (SUM): Training, Procedural, and Development Documentation", 31 Dec 2008.

"Software Product Specification", Executable Software, Source Files and Packaging Requirements, 31 Dec 2008.

"System Security Plan", Resources for the DNSSEC Initiative (RDI), 20 Jan 2011.

"IT Contingency Plan", Resources for the DNSSEC Initiative (RDI), 24 Feb 2011.

"Configuration Management Plan", Resources for the DNSSEC Initiative (RDI), 22 Feb 2011.

"RDI Configuration and Maintenance Guide", Resources for the DNSSEC Initiative (RDI), 17 Feb 2011.

"Software Product Specification", Executable Software, Source Files and Packaging Requirements, 30 March 2011.

"Software User Manual (SUM): Training, Procedural, and Development Documentation", 30 March 2011.

"DNSSEC Protocols and Standards Advancement", Technical Information Report, 30 March 2011.

"DNSSEC Awareness in Commonly Available Software Components", Technical Information Report, 30 March 2011.

8.2 Publications

[CATCH] Suresh Krishnaswamy, Wes Hardaker, Russ Mundy, "DNSSEC in Practice: Using DNSSEC-Tools to Deploy DNSSEC," catch, pp.3-15, Cybersecurity Applications & Technology Conference for Homeland Security, 2009

[SATIN] Wes Hardaker, Suresh Krishnawamy, “Enabling DNSSEC in Open-Source Applicatins”, 23 Jan 2011. To Appear in the Proceedings of Securing and Trusting Internet Names, SATIN 2011.

8.3 Web References

[DNSSEC-TOOLS] http://www.dnssec-tools.org

[DNSSEC-DEPLOY] http://www.dnssec-deployment.org

[SFT-TRK] https://www.dnssec-deployment.org/wiki/index.php/Tools_and_Resources

[DT-SBS] http://www.dnssec-tools.org/docs/step-by-step/sbs.pdf

[CRIB-SHEET] http://wordpress.test.dnssec-deployment.org/wp-content/uploads/2009/09/Setting-the-Parameters-2009112403.pdf

[GRAPHVIZ] http://www.graphviz.org/

[ISC_BIND] http://www.isc.org/software/bind

[TETHER] http://www.ethereal.com/

[LOGWATCH] http://www.logwatch.org/

[SRCFRG] http://sourceforge.net/

9 List of Symbols, Abbreviations and Acronyms

DNS	Domain Name System
DNSSEC	Domain Name System Security
ZSK	Zone Signing Key
KSK	Key Signing Key
ccTLD	Country Code Top-Level Domain
NSEC	Next SECure Record
NSEC3	NSEC record using one-way hashes instead of actual names
TAR	Trust Anchor Repositories
RFC	Request For Comments
DLV	Dynamic Look-Aside Validation
SHA	Secure Hash Algorithm
API	Application Programming Interface
DoD	Department of Defense
BIND	Berkeley Internet Name Daemon
FIPS	Federal Information Processing Standards
ROB	Rules of Behavior
RDI	Resources for the DNSSEC Initiative System (the C&A system)
DHS S&T	Department of Homeland Security, Science & Technology
IETF	Internet Engineering Task Force
DS	Delegation Signer
RR	Resource Records
PIR	Public Interest Registry
NIST	National Institute of Standards and Technology
TTL	Time To Live
SRCFRG	Source Forge